IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 2, 2019, accepted August 23, 2019, date of publication August 27, 2019, date of current version September 11, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2937809

Conditional Generative Adversarial Network-Based Data Augmentation for Enhancement of Iris Recognition Accuracy

MIN BEOM LEE, YU HWAN KIM, AND KANG RYOUNG PARK
Division of Electronics and Electrical Engineering, Dongguk University, Seoul 04620, South Korea

Corresponding author: Kang Ryoung Park (parkgr@dongguk.edu)

This work was supported in part by the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT through the Basic Science Research Program under Grant NRF-2019R1F1A1041123, in part by the NRF funded by the Ministry of Education through the Basic Science Research Program under Grant NRF-2018R1D1A1B07041921, and in part by the NRF funded by the Ministry of Education through the Basic Science Research Program under Grant NRF-2017R1D1A1B03028417.

ABSTRACT Presently, lots of previous studies on biometrics employ convolutional neural networks (CNN) which requires a large amount of labeled training data. However, biometric data are considered as important personal information, and it is difficult to obtain large amounts of data due to individual privacy issues. Training with a small amount of data is a major cause of overfitting and low testing accuracy. To resolve this problem, previous studies have performed data augmentation that are based on geometric transforms and the adjustment of image brightness. Nevertheless, the data created by these methods have high correlation with the original data, and they cannot adequately reflect individual diversities. To resolve this problem, this study proposes iris image augmentation based on a conditional generative adversarial network (cGAN), as well as a method for improving recognition performance that uses this augmentation method. In our method, normalized iris images that are generated through arbitrary changes in the iris and pupil coordinates are used as input in the cGAN-based model to generate iris images. Due to the limitations of the cGAN model, data augmentation, which uses the periocular region, was found to fail with regard to the improvement of performance. Based on this information, only the iris region was used as input for the cGAN model. The augmentation method proposed in this paper was tested using NICE.II training dataset (selected from UBIRS.v2), MICHE database, and CASIA-Iris-Distance database. The results showed that the recognition performance was improved compared to existing studies.

INDEX TERMS Biometric technology, iris recognition, deep learning, data augmentation, conditional generative adversarial network.

I. INTRODUCTION

Over the last decade, deep learning technology has achieved excellent performance in a variety of fields in computer vision, such as image classification and object detection. Zhang et al. proposed dual model learning combined with multiple feature selection for accurate visual tracking by fusing the handcrafted features with the multi-layer features extracted from the convolutional neural network (CNN) [69]. In other research, they proposed the method of spatially attentive visual tracking using multi-model adaptive response fusion [70]. In many cases, CNN models require a considerably large amount of data to be trained effectively. Training that uses insufficient data shows high classification performance with regards to the training dataset, but overfitting issues do occur, and classification performance is poor with regard to the testing dataset. To resolve the overfitting issues, techniques such as dropout [1] and batch normalization [2] were developed. In addition, researchers have used methods that create additional data by applying various geometric transformations to the existing training data. These methods are very useful. However, applying typical geometric transform-based data augmentation methods to small data sets, rather than larger datasets like ImageNet [4], is not sufficient for resolving these issues, because it produces very limited diversities in the existing data [3], [5].

The associate editor coordinating the review of this article and approving it for publication was Michele Nappi.

Therefore, there is a need for new methods to resolve the data scarcity problem.

Biometrics has evolved along with the development of pattern recognition research. Moreover, studies are being conducted that apply deep learning technology to biometrics [6], [7]. However, deep learning technology requires a larger dataset for training, and the datasets are very small in the case of data containing individual iris images for biometrics. Because of this problem, it is difficult to adequately train deep learning models. To resolve this problem, this study proposes iris image augmentation based on a conditional generative adversarial network (cGAN), as well as a method for improving recognition performance that uses this augmentation method. In our method, normalized iris images that are generated through arbitrary changes in the iris and pupil coordinates are used as input in the cGAN-based model to generate iris images. Due to the limitations of the cGAN model, data augmentation, which uses the periocular region, was found to fail with regard to the improvement of performance. Based on this information, only the iris region was used as input for the cGAN model.

This paper is organized as follows. Section II describes existing iris recognition research studies, and Section III describes the contributions of this study. Section IV describes the cGAN-based iris data augmentation that is proposed by this study and an iris recognition method, which uses this augmentation method. Section V presents experimental results with analysis, and Section VI presents the conclusions of this paper.

II. RELATED WORKS

A. DATA AUGMENTATION WITH VARIOUS BIOMETRIC DATA

Data augmentation is a technique that is necessary for deep learning technology, because overfitting occurs in training that is based on supervised learning using an inadequate dataset. Effective data augmentation methods reduce intra-class distances and increase inter-class distances to help improve performance [16]. Typical data augmentation techniques that have been used in existing deep learning studies include random translations, rotations, flips, the addition of Gaussian noise, random cropping, horizontal/vertical shifting, and zooming in/out [3], [16]. As explained in Section I, these geometric transform-based data augmentation methods lead to very limited diversities on existing data, and therefore are not adequate for resolving the problem of performance reductions that occurs due to the data scarcity problem, which includes overfitting [3], [5]. In order to solve this problem, studies are currently being conducted on performing data augmentation by using deep learning technology to generate images that are similar to the training dataset. This study was performed using the GAN [8] structure proposed by Goodfellow et al. Radford et al. proposed architectural guidelines focused on improving the training stability of conventional GAN with the perceptual quality of its generated images, and they proposed deep convolutional GAN (DCGAN) [9]. Minaee et al. proposed Finger-GAN, which reflects total variation (TV) in gradient updates to generate images with strengthened fingerprint connectivity in the generator of a model based on DCGAN [15]. Wang et al. proposed a method that incorporates structural similarity (SSIM) loss in order to prevent overfitting in DCGAN-based models for palmprint data augmentation [16].

B. DATA AUGMENTATION WITH IRIS DATA

1) DATA AUGMENTATION BY NON-DEEP FEATURE-BASED METHOD

Because the target of our study was iris recognition, this section focuses on the existing studies concerned with this topic. Existing iris recognition studies are generally divided into handcrafted feature-based methods [27]–[30], [45]–[51], [53], and deep feature-based methods [31]–[35], [52], [54], depending on the method by which features are extracted from iris images.

A large amount of labeled training data is needed to train the CNN models for the deep feature-based methods. However, the images used on biometrics contain important personal information, and therefore it is difficult to obtain a large amount of data due to personal privacy issues. Training that uses a small amount of training data is a major cause of overfitting and low testing accuracy. To resolve this, previous studies have used methods that perform geometric transform-based or brightness change-based data augmentation to increase the number of training data. To extract compact and discriminative features, Zhang et al. proposed a Maxout CNNs model that uses maxout units, which are more efficient than rectified linear units (ReLU), generally used as an activation function. In the training process, they performed data augmentation with a method that randomly crops the training dataset. They also proposed a method that performs matching by calculating the cosine distance between the recognition images and the enrolled images using feature vectors obtained by inputting normalized iris and periocular images in the Maxout CNNs model [36]. Xu et al. proposed a segmentation network for accurate iris recognition. They also proposed an iris recognition method that performs data augmentation on the training dataset by mirroring and adjusting the brightness of the original images and trains a Siamese network, which is based on ResNet-18 to perform classification [37]. Zanlorensi et al. proposed a method, which uses in-plane rotation to perform data augmentation on the training dataset and then uses ResNet-50 or a visual geometry group (VGG) model to perform recognition [38]. Lee et al. proposed a method that adjusts the center coordinates of the iris and pupil and performs data augmentation based on imaging translation and cropping. The method extracts features from three CNN networks and measures their Euclidean distance, subsequently performing a score fusion via the weighted product method [6]. Although it is not the study on iris recognition, Zhang et al. proposed

TABLE 1. Comparison of existing and proposed methods. (d', EER, and GAR mean d-prime value, equal error rate, and genuine acceptance rate (100 - false rejection rate (FRR)), respectively. A detailed explanation about this is included in Sections IV.*D* and V.*D*.) (A: Noisy iris challenge evaluation-part II (NICE.II) training dataset / B: Institute of automation of Chinese academy of sciences (CAISA)-Iris-Ver.1.0 / C: Mobile iris challenge evaluation (MICHE) II competition database / D: CASIA-Iris-Distance / E: Notre Dame (ND)-Iris-0405 / F: CASIA-Iris-Thousand / G: CASIA-Iris-Mobile-Ver.1.0 / H: CASIA-Iris-Interval / I: MICHE database / J: Indian Institute of Technology (IIT) Delhi database / K: Subset of University of Beira iris (UBIRIS).v2 database).

Category	Augmentation method		Recognition method	Accuracy	Pros	Cons
Without data augmentation			Method that classifies via K-NN classifier [27]	d' of 2.82, EER of 10% (K)	Little time required to develop the algorithm	Lower recognition performance than deep feature-based methods with data augmentation
			Using reverse biorthogonal wavelet transform [45]	d' of 1.09 (A)		
			RANSAC-based non-circular iris boundary detection and recognition [46]	d' of 1.32 (A)		
			Fusion of LBP and BLOBs features [47]	d' of 1.48 (A)		
			WCPH-based representation of local texture pattern [48]	d' of 1.58 (A)		
			CLAHE-based image enhancement [49]	EER of 18.82% (A)		
			Pre-classification based on both eyes and eye color [50]	EER of 16.94%, d' of 1.64 (A)		
			Using LBP-based periocular information along with iris recognition [51]	EER of 18.48%, d' of 1.74 (A)		
			Combining color and shape descriptors [53]	EER of about 16%, d' of about 2.42 (A)		
			Method that extracts features via Fourier descriptors [28]	EER of 0.17% (B)		
			Method that uses 1-D Log-Gabor filter and MB-TLBP [29]	EER of 1.22% (C)		
			Score fusion of distances found by color, texture, and cluster descriptors [30]	EER of 29% (C)		
			Adaboost training using multi-orientation 2D Gabor-based feature set [52]	d' of 2.28 (A)		
			Score fusion by periocular matching score, SOBoost, and diffusion distance [54]	d' of 2.57 (A)		
			Small and local pattern extraction using FCN-based model, and ETL [31]	EER of 3.85% (D)		
			DeepIrisNet [32]	EER of 2.19% (E)		
			Using an SVM to classify features extracted by AlexNet [33]	GAR of 98% (B), GAR of 98% (F), GAR of 89% (H)		
			Using an SVM to classify features extracted by VGG-Net 16 [34]	GAR of 99.4% (J)		
			Five state-of-the-art and off-the-shelf CNNs [35]	GAR of 98.5% (F)		
With data augmentation	Geometric transform or bright change-based	Random cropping	Maxout CNNs [36]	EER of 0.6% (G)	Resolves the problem of overfitting in regards to training data	Generated data has high correlation with original data in addition to insufficient data diversities
		Mirroring and adjusting the image brightness	ResNet-18-based Siamese model [37]	EER of about 3.43% (H)		
		In-plane rotation	ResNet-50 and VGG [38]	d' of 2.2480, EER of 13.98% (A)		
		Image translation and cropping by adjusting the iris and pupil centers	Three CNNs-based method [6]	d' of 2.62, EER of 10.36% (A) d' of 1.87 ~ 2.26, EER of 16.25 ~ 17.9% (I)		
	GAN-based	Iris-GAN-based	Iris recognition was not performed [19]	Recognition accuracy was not measured	Produces sufficient diversities of training data	Long training time required
		cGAN-based (**Proposed method**)	Three CNNs and SVM-based score fusion	d' of 3.31, EER of 8.58% (A) d' of 1.87 ~ 2.09, EER of 15.49 ~ 17.39% (I) d' of 3.29, EER of 2.96% (D)		

a fully convolutional network (FCN) based on DenseNet for classifying remote sensing scene. In order to avoid overfitting of their CNN, they performed data augmentation for training data by brightness, color, contrast, sharpness, and rotation (of random angles) transformation [68].

2) DATA AUGMENTATION BY DEEP FEATURE-BASED METHOD

However, training data, which is created in this way, has a high correlation with the original data, and therefore it is unable to produce sufficient diversities on the biometric data. To resolve this problem, deep learning-based data augmentation is currently being studied in a variety of fields such as leaf counting, plant phenotyping, and liver lesion classification [5], [12], [14]. Minaee et al. proposed Iris-GAN, which uses a deep convolutional GAN (DCGAN) model to augment iris images captured in an NIR environment [19]. However, the DCGAN used in that study generates iris images from D-dimensional noise vectors. Therefore, there are unrealistic portions in the generated iris images (especially around the iris boundary), which are different from actual iris images. Moreover, that study only used the Frechet Inception Distance (FID) to measure the similarity between the distribution of the generated and actual iris images, and it did not produce experimental results in which the iris recognition performance was improved by performing actual training with the generated iris images.

To resolve the problem of these previous studies, this paper proposes cGAN-based iris image augmentation and a method of improving recognition performance that uses this augmentation method.

Table 1 below summarizes and compares the pros and cons of the methods proposed in this and the previous studies.

III. CONTRIBUTIONS

Our research is novel in the following four ways compared to previous studies.

- There is no previous research of iris recognition which uses training data augmented by cGAN, and most of them augmented the data by geometric transform and the adjustment of image brightness. Therefore, this is the first study to improve iris recognition performance by using cGAN to augment training data and training a CNN with the augmented data.
- Due to the limitations of the cGAN model, data augmentation, which uses the periocular region, was found to fail with regard to the improvement of performance. Based on this information, only the iris region was used as input for the cGAN model.

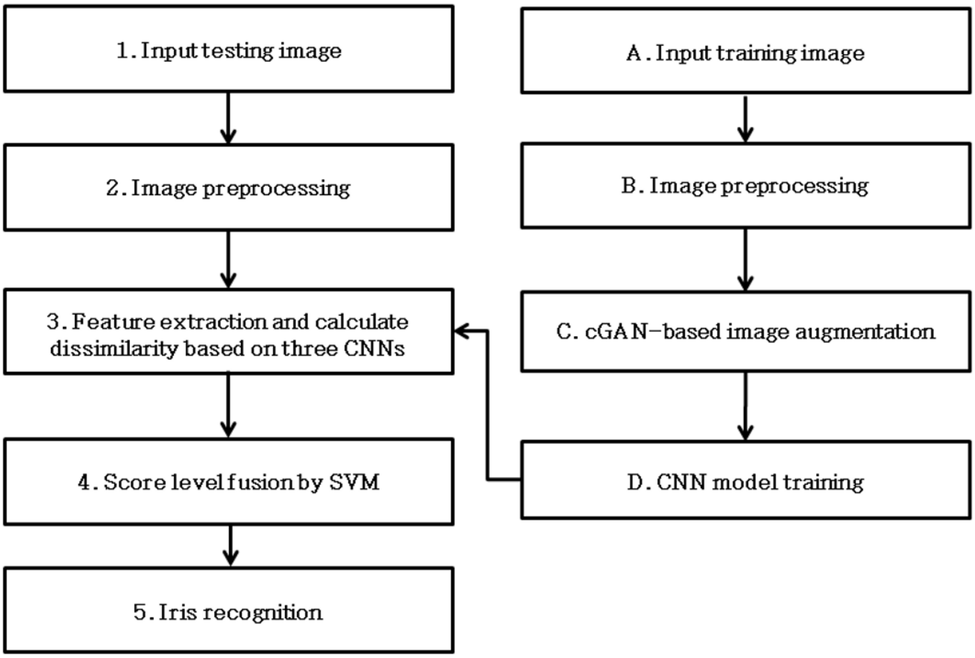

1. Input testing image
2. Image preprocessing
3. Feature extraction and calculate dissimilarity based on three CNNs
4. Score level fusion by SVM
5. Iris recognition
A. Input training image
B. Image preprocessing
C. cGAN–based image augmentation
D. CNN model training

FIGURE 1. Overview of the proposed method.

- This study employs a data augmentation method that is suitable for iris recognition. Instead of using whole eye or iris image in Cartesian coordinate, the normalized iris images of polar coordinate that are generated through arbitrary changes in the iris and pupil coordinates are used as input in the cGAN-based model to generate iris images.
- In order to make fair comparisons to studies performed by other researchers, we constructed our trained CNN models and generated iris images public through [59].

IV. PROPOSED METHOD

A. OVERVIEW OF PROPOSED METHOD

Fig. 1 shows an overall flowchart of the algorithm proposed in this study. The iris regions are detected in the input training images to generate normalized iris images (step (B) of Fig. 1). The normalized iris images are used as input in the cGAN-based model to train the generators and discriminators, and data augmentation is performed (step (C) of Fig. 1). The augmented dataset and the training images are used to train the CNN model for extracting features (step (D) of Fig. 1). In the testing process, the iris and periocular regions are detected in the input testing images, and normalized iris and periocular images are generated (step (2) of Fig. 1). The generated normalized iris and periocular images are entered as input in the three CNN models that were trained in step (D) of Fig. 1, and features are extracted. The dissimilarity (distance) between these features and the enrolled features is calculated (step (3) of Fig. 1). A support vector machine (SVM) is used to perform score level fusion on the three calculated dissimilarities (distances) of the iris and periocular regions to find a single score (step (4) of Fig. 1), and this score is used to perform iris recognition (accept as genuine matching or reject as imposter matching) (step (5) of Fig. 1). More detailed information on this method is discussed in each of the sub-sections below.

B. IMAGE PREPROCESSING

In this study, the following two circular edge detector (CED) were used to extract the iris region [6], [50].

$$\underset{(x_0,y_0,r)}{\arg\max}\left[\frac{\partial}{\partial r}\left(\int_{-\frac{\pi}{4}}^{\frac{\pi}{6}}\frac{I\left(x,y\right)}{5\pi r/12}ds+\int_{\frac{5\pi}{6}}^{\frac{5\pi}{4}}\frac{I\left(x,y\right)}{5\pi r/12}ds\right)\right] \quad (1)$$

where r is the radius of iris region. The coordinates (x_0, y_0) denote the center position of the iris region. Fig. 2(b) shows the result of the detected iris region.

In addition to the iris region detected in Fig. 2(b), this study also used images that include the periocular region around the iris region, which extends the iris radius (*IRrad*) detected by Equation (1) based on the detected iris center location. These images were used as input for the CNN to perform recognition [6]. That is, the areas specified by $w_1 \times IRrad$ and $w_2 \times IRrad$, as shown in Figs. 3(b) and (c), were detected and used as input for the CNN [6]. Then, this study performed the size normalization process shown in Fig. 4 on the iris and periocular images obtained from Fig. 3 [6]. In this method, the iris images of polar coordinates are divided into 8 tracks and 256 sectors, as shown in Fig. 4(b). In each track, the pixel values are averaged in the vertical (ρ axis) direction by using a one-dimensional (1-D) Gaussian kernel. Consequently, the normalized iris and periocular images of 256 × 8 pixels are produced.

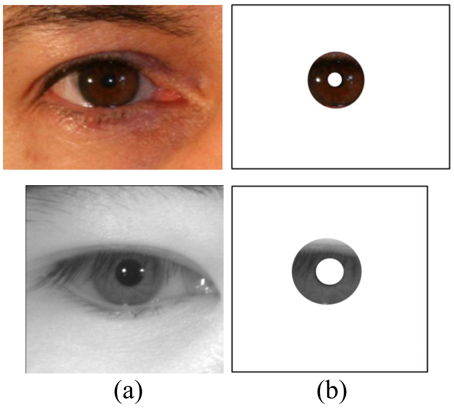


FIGURE 2. Examples of iris region detection. (a) Original image, (b) Detected iris region. The 1st and 2nd row images are the examples from NICE.II training and CASIA-Iris-Distance databases, respectively.

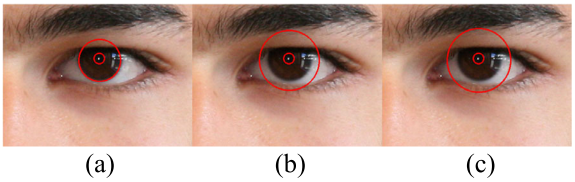


FIGURE 3. Examples of iris and periocular regions. (a) Iris region. (b) Periocular region based on $w_1 \times$IRrad. (c) Periocular region based on $w_2 \times$IRrad.

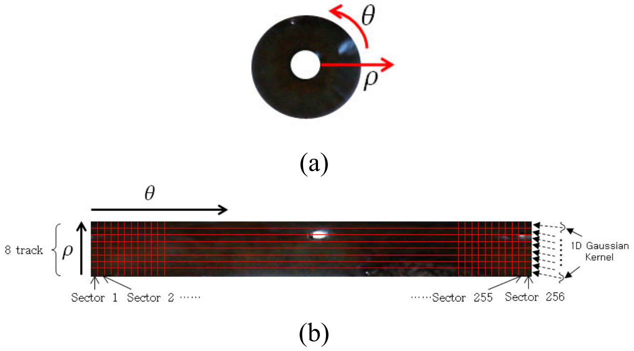


FIGURE 4. Example of the normalized iris image. (a) Iris region in Cartesian coordinates. (b) Normalized iris image of (a) in polar coordinates.

C. cGAN-BASED AUGMENTATION OF TRAINING DATA AND CNN TRAINING

Generally, the images used for biometrics are considered important personal information, and it is difficult to obtain a large number of images from various people. The ideal features for excellent iris recognition performance should maintain invariance between data items in the same classes, and they should have highly distinctive characteristics between data items in different classes. To extract these features in a CNN model, the model must be sufficiently trained with a large number of iris images that show the individuals' unique features.

Previous studies on CNN-based classification increased the number of data in insufficient training datasets and prevented overfitting in the training data by performing geometric transform-based data augmentation, including random translations, rotations, flips, adding Gaussian noise, random cropping, horizontal/vertical shifting, and zooming in/out [3], [16]. However, in the case of iris patterns, it is difficult to use simple geometric augmentation methods, such as mirroring, because the position information of pattern is important. In addition, the data generated by such methods has a high correlation with the original data, and therefore it cannot produce sufficient diversities on the iris data. Furthermore, a variety of noise such as optical blur, motion blur, off-angle views, and specular reflections (SR) are included in the NICE.II training dataset [13] and the MICHE database [26] that are used in this study. Therefore, it is difficult to obtain high recognition performance with the testing dataset. This study considers these problems comprehensively and proposes cGAN-based training data augmentation, as well as a method for improving the performance of recognition, which uses the augmented data.

$$\mathcal{L}_{GANs}\left(G,D\right)=\mathbb{E}_y[\log D(y)]+\mathbb{E}_z[\log(1-D\left(G(z)\right)] \quad (2)$$

GAN is a deep learning architecture for generating images, and it is composed of the generator and discriminator, which were proposed by Goodfellow *et al.* [8]. The generator of GAN generates fake images (($G(z)$ in Fig. 5) from the input noise z. The fake images created by the generator are given to the discriminator, which attempts to perform a binary classification to discern whether the images received as input are genuine images or fake images created by the generator. The discriminator is trained to maximize the probability where it correctly discerns the real images and the fake images created by the generator. At the same time, the generator is trained to minimize $\log\left(1-D\left(G\left(z\right)\right)\right)$ in Equation (2) [8]. Through this competitive and repetitive training, the generator is able to generate fake images that are similar to real images. However, because it generates images based on input noise z, it is difficult to control the images created by the generator, and it experiences difficulties when generating high-resolution images. Because of this problem, Mirza et al. proposed cGAN [24], which uses both the input noise z and the extra information $\boldsymbol{x}$ to generate data as shown in Fig. 6.

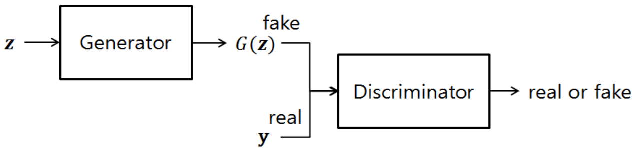


FIGURE 5. Structural concept of GAN.

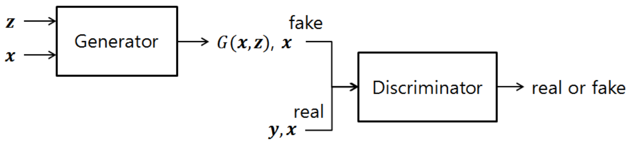


FIGURE 6. Structural concept of conditional GAN.

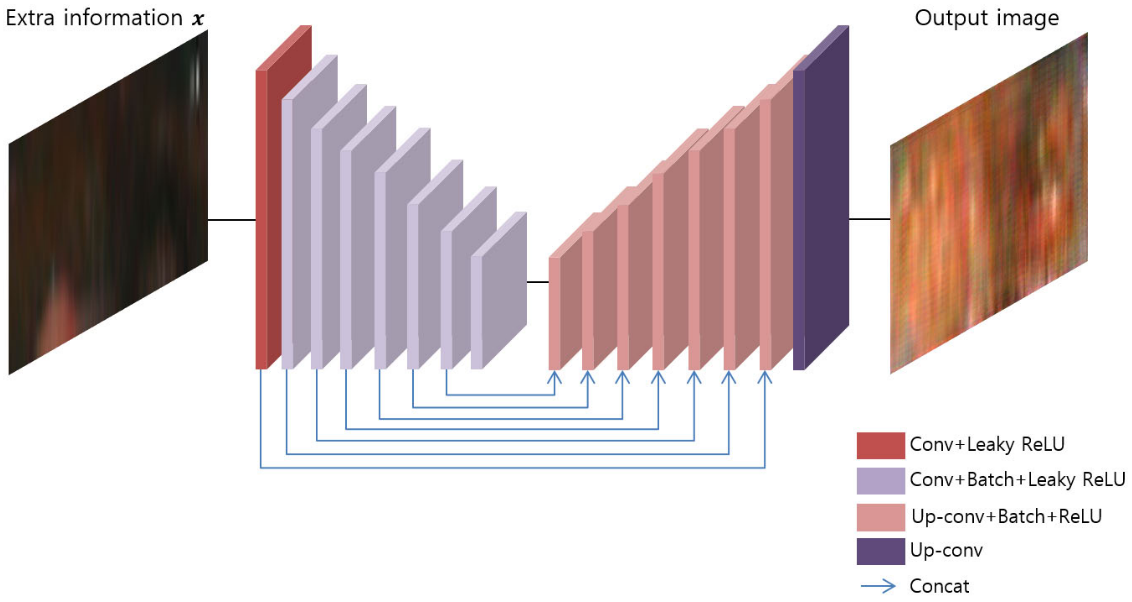


FIGURE 7. Generator of pix2pix GAN model.

In other words, cGAN adds extra information to the generator to control data generation.

$$\mathcal{L}_{cGAN}\left(G,D\right)=\mathbb{E}_{\boldsymbol{x},\boldsymbol{y}}[\log D(\boldsymbol{x},\boldsymbol{y})]+\mathbb{E}_{\boldsymbol{x},z}[\log(1-D(\boldsymbol{x},G(\boldsymbol{x},z))] \tag{3}$$

Equation (3) is the objective function of cGAN. The generator and discriminator of cGAN are conditioned on some extra information $\boldsymbol{x}$. $\boldsymbol{x}$ can be any kind of auxiliary information, such as class labels or data from other modalities [24]. The pix2pix GAN model proposed in the study by Cheng *et al.* [10] was used as the cGAN structure of the method in this study. The pix2pix GAN model has been widely used in various studies in the field of image-to-image translation. U-Net [11] was used as the generator in this model, and a skip-connection was added between layer *i* and layer *n-i* when the total number of layers in the encoder-decoder structure was *n*, to avoid losing low-level information that is reduced by progressive down-sampling. This method is effective in preserving the low-level information of the input in the output of generator [11]. Furthermore, when the generator is trained, the L1 distance (Equation (4)) between the ground truth data and the data created by the generator is reflected in the objective function, as shown in Equation (5). Thereby, low-level information is strengthened so that images can be generated that are clearer than conventional cGAN [10]. Moreover, to prevent blurring in the images created by the generator, the PatchGAN concept was applied. This is a method that attempt to classify the input images of discriminator in $N\times$ area patches [10].

$$\mathcal{L}_{L1}\left(G\right)=\mathbb{E}_{\boldsymbol{x},y,z}\left[\|\boldsymbol{y}-G\left(\boldsymbol{x},z\right)\|_1\right] \tag{4}$$

$$G^*=arg\min_G\max_D \mathcal{L}_{cGAN}+\lambda\mathcal{L}_{L1}\left(G\right) \tag{5}$$

Fig. 7 and Table 2 show the structure of the generator of the pix2pix GAN model used for augmentation in this study.

The generator of the pix2pix GAN model receives images of the size 256 × 256 × 3 (height × width × channel) as input, and the feature maps are calculated with a 5 × 5 filter in the 1^{st} - 8^{th} convolutional layers of encoder. This was designed so that a separate pooling layer, such as max pooling, is not used, and the padding and stride are calculated as 2 × 2 to reduce the size of the feature map. The feature map, which is reduced to 1 × 1 × 512 by the 8^{th} convolutional layer, is up-sampled through the 1^{st}–8^{th} transposed convolutional layers of decoder. To avoid losing low-level information, this process adds a skip-connection between layer *i* and layer *n-i*when the total number of layers in the generator model is *n* (concatenated explained in Table 2).

Table 3 shows the structure of the discriminator model, which is used to discern whether the images created by our generator are genuine or fake. The input for the discriminator is a pair of images consisting of an image created by the generator and an image that uses extra information (fake image), or a pair of images consisting of a geometric center image and an image that uses extra information (real image), as shown in Equation (3). The input image is reduced to a 32 × 32 × 512 size by the 4 convolutional layers. The reduced feature map is produced as the final value regarding whether the image is real or fake via the linear regression and sigmoid function of fully connected layer.

To augment the iris images with the pix2pix GAN model, this study performed the task of creating a training dataset with a method of image translation and cropping by adjusting the coordinates of the iris and pupil centers. That is, the training dataset was created by selecting the geometric center image of each class and then artificially moving the x and y positions of the center by ±4 positions horizontally and ±4 positions vertically, based on the iris center and pupil center.

TABLE 2. Generator model used in our research.

Layer type		Number of filters	Size of feature map (height×width×channel)	Filter size (height×width×channel)	Number of stride (height×width)	Number of padding (height×width)
	Image input layer		256×256×3			
Encoder	1st convolutional layer	64	128×128×64	5×5×3	2×2	2×2
	Leaky ReLU layer		128×128×64			
	2nd convolutional layer	128	64×64×128	5×5×64	2×2	2×2
	Batch normalization		64×64×128			
	Leaky ReLU layer		64×64×128			
	3rd convolutional layer	256	32×32×256	5×5×128	2×2	2×2
	Batch normalization		32×32×256			
	Leaky ReLU layer		32×32×256			
	4th convolutional layer	512	16×16×512	5×5×256	2×2	2×2
	Batch normalization		16×16×512			
	Leaky ReLU layer		16×16×512			
	5th convolutional layer	512	8×8×512	5×5×512	2×2	2×2
	Batch normalization		8×8×512			
	Leaky ReLU layer		8×8×512			
	6th convolutional layer	512	4×4×512	5×5×512	2×2	2×2
	Batch normalization		4×4×512			
	Leaky ReLU layer		4×4×512			
	7th convolutional layer	512	2×2×512	5×5×512	2×2	2×2
	Batch normalization		2×2×512			
	Leaky ReLU layer		2×2×512			
	8th convolutional layer	512	1×1×512	5×5×512	2×2	2×2
	Batch normalization		1×1×512			
	ReLU layer		1×1×512			
Decoder	1st up-conv layer	512	2×2×512	5×5×512	2×2	
	Concatenates		2×2×1024			
	ReLU layer		2×2×1024			
	2nd up-conv layer	512	4×4×512	5×5×1024	2×2	
	Concatenates		4×4×1024			
	ReLU layer		4×4×1024			
	3rd up-conv layer	512	8×8×512	5×5×1024	2×2	
	Concatenates		8×8×1024			
	ReLU layer		8×8×1024			
	4th up-conv layer	512	16×16×512	5×5×1024	2×2	
	Concatenates		16×16×1024			
	ReLU layer		16×16×1024			
	5th up-conv layer	256	32×32×256	5×5×1024	2×2	
	Concatenates		32×32×512			
	ReLU layer		32×32×512			
	6th up-conv layer	128	64×64×128	5×5×512	2×2	
	Concatenates		64×64×256			
	ReLU layer		64×64×256			
	7th up-conv layer	64	128×128×64	5×5×256	2×2	
	Concatenates		128×128×128			
	ReLU layer		128×128×128			
	8th up-conv layer (output layer)	3	256×256×3	5×5×128	2×2	

TABLE 3. Discriminator model used in our research.

Layer type	Number of filters	Size of feature map (height × width × channel)	Filter size (height × width × channel)	Number of stride (height × width)	Number of padding (height × width)
Image input layer		256×256×6			
1st convolutional layer Leaky ReLU layer	64	128×128×64	5×5×6	2×2	2×2
2nd convolutional layer Batch normalization Leaky ReLU layer	128	64×64×128	5×5×64	2×2	2×2
3rd convolutional layer Batch normalization Leaky ReLU layer	256	32×32×256	5×5×128	2×2	2×2
4th convolutional layer Batch normalization Leaky ReLU layer	512	32×32×512	5×5×256	1×1	2×2
Linear regression		1			
Sigmoid layer (output layer)		1			

Subsequently, the image is cropped to increase the existing original dataset by a factor of 81 (9 × 9). The generated training data was entered as extra information in pix2pix GAN, as shown in Fig. 7. The geometric center image of each class was provided as a ground truth image to train the model. In this study, the image with the lowest average of the CNN feature-based distances to the images in a class was selected as the geometric center image of the class. If the training dataset used in training is input to the completely trained generator of test dataset model, new training images are obtained, which are created to be close to the ground truth. This study combined the data that was generated in this way and the original training data to train the three CNN models for iris recognition, which are shown in Fig. 8. These three CNN models for iris recognition were used with reference to a previous study [6]. As described in Section III, the data created by pix2pix GAN and the original 81 times augmented training data was used only in the training of the 1st CNN in Fig. 8. In the training of the 2nd and 3rd CNN, only the original 81 times augmented training data was used. This is because in the case of the iris region used in the 1st CNN, the iris data was successfully generated by pix2pix GAN, such that it was close to the real data. However, in the case of the periocular region used in the 2nd and 3rd CNN, more high frequency components such as eyelash, eyelid, or double eyelid areas were included than in the iris pattern, as shown in Fig. 3. Due to the characteristics of pix2pix GAN, these high frequency components were not generated to be close to the real data.

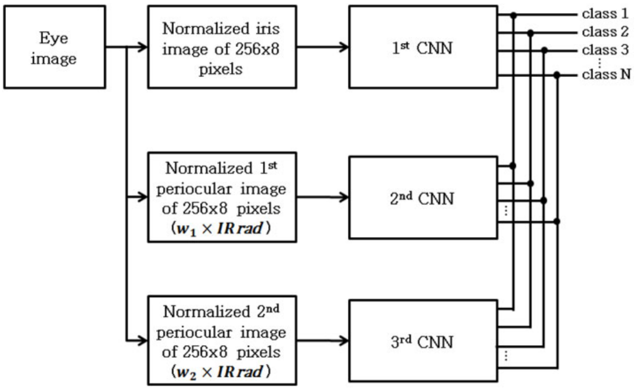

Eye image
Normalized iris image of 256x8 pixels
1st CNN
class 1
class 2
class 3
class N
Normalized 1st periocular image of 256x8 pixels
$(w_1 \times IR\,rad)$
2nd CNN
Normalized 2nd periocular image of 256x8 pixels
$(w_2 \times IR\,rad)$
3rd CNN

FIGURE 8. Iris recognition using three CNNs.

D. FEATURE EXTRACTION USING THREE CNN MODELS AND IRIS MATCHING

In this study, three CNN models were used to perform feature extraction for iris recognition [6], as shown in Fig. 8. 4096-dimensional features were extracted in each 1st fully connected layer of the three CNNs. Then the Euclidean distances between the three pairs of 4096-dimensional features between enrolled and recognized images were found to obtain the 3 distances [6].

Different from previous research using weighted SUM and weighted PRODUCT rules for score level fusion [6], in this study, the score level fusion was performed by using an SVM on the three distances. An SVM is an efficient classification method, based on the use of support vectors [20]–[22], [60]. This method maximizes the distance (or margin) between the classes and creates an optimized hyper-plane. In the case of complex problems, such as non-linear environments rather than simple classification problems, a kernel function is used to transform low dimension space data into higher dimension space data to make it easier for the hyper-plane to perform classification. SVM constructs a hyper-plane by selecting several support vectors, as shown in Equation (6). In this equation, $\mathrm{x_i}$ and $\mathrm{y_i}$ are the selected support vectors and their corresponding labels (–1 or 1). a_i and b are the parameters of the SVM model, and $K(\cdot)$ is the kernel function. In our experiments, we experimentally selected a radial basis function (RBF) of Equation (7) as optimal one.

$$f(x) = sgn(\sum\nolimits_{\mathrm{i}=1}^{\mathrm{k}} a_i y_i K(x, x_i) + b) \quad (6)$$

$$K(x_i, x_j) = e^{-\gamma \|x_i - x_j\|^2} \quad (\gamma > 0) \quad (7)$$

If the score (distance) produced by the SVM is greater than the threshold, the case is considered as imposter matching. If it is less than the threshold, authentic (genuine) matching is considered to have occurred. There is generally a trade-off relationship between the two errors, which occurs in this situation, i.e., the false acceptance rate (FAR) (the error of accepting imposter matching as authentic matching) and the false rejection rate (FRR) (the error of rejecting authentic matching as imposter matching). The error rate at the threshold where FAR and FRR are the same is known as the EER.

V. EXPERIMENTAL RESULTS

A. EXPERIMENTAL DATASETS

To evaluate the performance of the method proposed in this study, experiments were performed using three open databases. The first database was the NICE.II training dataset, which is a part of the UBIRIS.v2 database that was selected to evaluate iris recognition performance in an unconstrained environment under very noisy visible light. This dataset was used to evaluate performance in the NICE.II contest [13]. The NICE.II training dataset includes 1000 eye images in 171 classes. The image resolution is 400 × 300 pixels, and a high-resolution visible light camera was used with visible light illumination. The iris images were captured from people walking at a distance of 4–8 m from the camera [25]. This images of dataset include many performance-reducing factors such as in-plane rotation, low-illumination, blurring, and off-angle views.

The second database, MICHE [26], was created for studies on iris recognition in a mobile device environment. These sub-datasets are organized by the type of mobile device used to capture the image such as Galaxy S4, Galaxy Tab2, and iPhone5. The images were captured by the front or rear cameras of smart devices in indoor and outdoor visible light environments. They include performance-reducing factors, such as optical and motion blur, off-angle views, and specular reflections.

To evaluate the applicability of the method proposed in this study in NIR camera and illuminator environments rather than just visible light camera environments like the first and second databases, the CASIA-Iris-Distance database [39] was used as the third database. This database is divided into the left and right eyes of 142 people for a total of 284 classes. In this study, 2068 images of the left and right eyes were used in the experiments, for a total of 4136 images and 284 classes. This database includes iris images captured by the self-developed long-range multi-modal biometric image acquisition and recognition system (LMBS). Detailed specifications and explanations of the physical system with magnification factor and focal length of the camera lens are not unveiled. In addition, in order to consider the various capturing environments along the long Z-distance (from a distance of 2.4 m to 3 m), various noise factors such as severe off-angle, specular reflection on glasses, low illumination, and hair occlusion were included. Therefore, the accuracy of iris recognition with CASIA-Iris-Distance is usually lower than the accuracies obtained with other NIR iris databases [31]. In this study, the number of classes in each database was divided in half to create A and B sub-datasets, and training and testing were performed based on a two-fold cross validation method. Hence, training was performed with one dataset, and subsequently testing was performed with the other sub-dataset (1st fold cross validation).

These two sub-datasets were switched, and training and testing were performed once more (2nd fold cross validation), upon which the average accuracy was measured. Table 4 shows the detailed description of 3 open databases, the training data, and the generated training data by GAN. Fig. 9 shows example images from the three datasets used in the experiments.

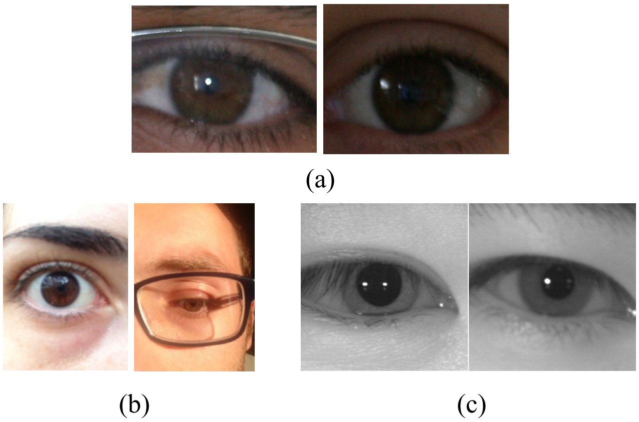


FIGURE 9. Example images from the datasets used in the experiments. (a) NICE.II training dataset. (b) MICHE database. (c) CASIA-Iris-Distance database.

B. TRAINING OF pix2pix GAN MODEL FOR DATA AUGMENTATION AND THREE CNN MODELS FOR RECOGNITION

The pix2pix GAN was implemented using the TensorFlow framework (version 1.12.0) [40]. For training, this study used an Adam optimizer, which is the method for first-order gradient-based optimization of stochastic objective functions by adaptive estimates of lower-order moments [42]. The initial parameters for this optimizer are a learning rate of 0.002, momentum of 0.5, momentum2 of 0.999, and

TABLE 4. Detailed descriptions of original and augmented datasets.

Dataset		Sub-dataset	Number of classes	Number of iris images in original dataset	Training dataset (A) for GAN training	Augmented dataset (A') generated in GAN	Dataset (A+A') for training 1st CNN in Fig. 8
NICE.II training		A sub-dataset	86	492	39852	39852	79704
		B sub-dataset	85	508	41148	41148	82296
MICHE	Galaxy S4	A sub-dataset	35	330	26730	26730	53460
		B sub-dataset	35	342	27702	27702	55404
	Galaxy Tab2	A sub-dataset	28	123	9963	9963	19926
		B sub-dataset	28	135	10935	10935	21870
	iPhone5	A sub-dataset	35	312	25272	25272	50544
		B sub-dataset	35	298	24138	24138	48276
CASIA-Iris-Distance		A sub-dataset	142	2068	167508	167508	335016
		B sub-dataset	142	2068	167508	167508	335016

epsilon of 1e–08. In the training process, the batch size was 16, and training was performed for 200 epochs.

Further, the three CNN models in Fig. 8 were implemented using the Caffe framework (version 1) [43], and the Adam optimizer was used. The initial parameters for this optimizer are a learning rate of 0.001, momentum of 0.9, momentum2 of 0.999, and epsilon of 1e–08. The detailed explanations of these parameters are provided in the study by Kingma *et al.* [42]. The convolution filter was initialized in a method suggested by He *et al.* [44], and the biases were initialized to zero. A batch size of 128 was used, and learning was conducted in 50 epochs. Fig. 10 shows the training loss and accuracy when the three CNN networks were trained by the A sub-dataset and the B sub-dataset of the NICE.II training dataset in Table 4. In the training results, all training losses converge close to 0, and the training accuracies converge close to 100%. Thereby, it can be assumed that the three CNN models used in this study were sufficiently trained.

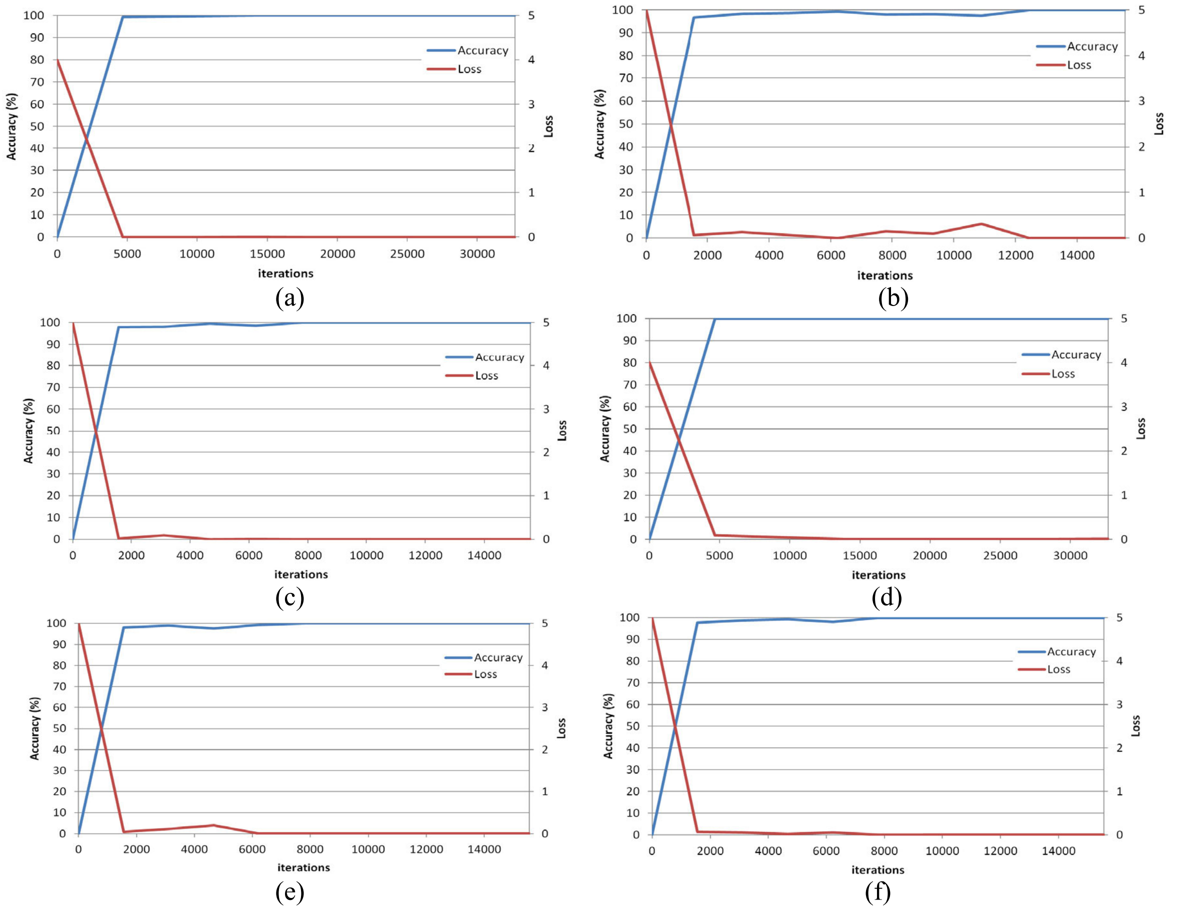
To enable a fair comparison with other existing methods, this study used the augmented data generated by the proposed method only in the training process. In the testing process, the non-augmented original data was used. The experiments were performed using an Intel ® Core™ i7-7700 CPU @ 3.6 GHz (4 cores) with 32 GB of main memory, and a NVIDIA GeForce GTX 1080 (2560 compute unified device architecture (CUDA) cores) [41] with a graphics memory of 8 GB (NVIDIA, Santa Clara, CA, USA).

C. IMAGE GENERATION BY pix2pix GAN MODEL

Fig. 11 shows examples of normalized iris and periocular images generated by the pix2pix GAN model that was used in this study, as depicted in Fig. 4(b). The generator was trained to receive the images in Fig. 11(b) as input and generate the images in Fig. 11(a). Fig. 11(c) shows the ultimately generated images of the fully trained generator. As shown in the images of the 1st and 2nd rows of Fig. 11, the generator refers to the input iris images (Fig. 11(b)) and generates iris images that have a color and texture that is similar to the input geometric center image of the iris (Fig. 11(a)), as shown in Fig. 11(c). However, in the images shown in the 3rd and 4th rows of Fig. 11, the generated periocular images have relatively lower similarity to the geometric center images than the cases of the 1st and 2nd rows. That is because in the case of the periocular region, more high frequency components such as eyelash, eyelid, or double eyelid areas are included than in the iris pattern, as shown in Fig. 3. Due to the characteristics of pix2pix GAN, these high frequency components are not generated to be close to the real data.

D. TESTING OF IRIS RECOGNITION WITH NICE.II TRAINING DATASET

In this study, the EER, receiver operating characteristic (ROC) curve, and decidability value (d-prime value) were used to evaluate the iris recognition performance. The d-prime value was used for objective iris recognition performance evaluations in the NICE.II contest [13], and it is calculated by Equation (8). The d-prime value is calculated using the mean (μ_A and μ_I) and standard deviations (σ_A and σ_I) of the authentic and imposter matching distributions. In biometrics, false acceptance cases and false rejection cases mainly occur due to overlap between the authentic and imposter matching distributions. Therefore, as these two distributions



(a)

(b)

(c)

(d)

(e)

(f)

FIGURE 10. Loss and accuracy curves of CNN training. Training of (a) the 1st CNN of Fig. 8 with A sub-dataset of Table 4, (b) the 2nd CNN of Fig. 8 with A sub-dataset of Table 4, (c) the 3rd CNN of Fig. 8 with A sub-dataset of Table 4, (d) 1st CNN of Fig. 8 with B sub-dataset of Table 4, (e) 2nd CNN of Fig. 8 with B sub-dataset of Table 4, (f) 3rd CNN of Fig. 8 with B sub-dataset of Table 4.

become farther apart so that they do not overlap, the FAR, FRR, and EER generally become smaller. The d-prime value of Equation (8) becomes larger as the two distributions grow farther apart, and it becomes smaller as the two distributions come closer, with a high degree of overlap. Therefore, as the d-prime value becomes larger, the performance of biometric system is judged to be better.

$$d' = \frac{|\mu_A - \mu_I|}{\sqrt{\frac{\sigma_A^2+\sigma_I^2}{2}}} \tag{8}$$

TABLE 5. Comparison of recognition accuracy when using training data generated by the pix2pix GAN model vs. not using this data.

Method	EER (%)	d-prime value
Recognition in case of training with A in Table 4	10.37	2.42
Recognition in case of training with A + A' in Table 4	16.23	2.02

In the first experiment, a recognition performance comparison was made between the case in which training data for the periocular regions, used as input in the 2nd CNN, was generated by the pix2pix GAN model used in this study (the case in Table 4 where A+A' was used) and the case in which this did not occur (the case in Table 4 where only A was used). As shown in Table 5, the recognition error (16.23%) of periocular region in case of using the augmented data by cGAN for training is higher than that (10.37%) in case of not using the augmented data by cGAN. Inclusion of the periocular region can obtain important elements for distinguishing people using the skin color, wrinkles and eyelids, however this is strongly influenced by illumination. Dark skin can be captured as light in bright illumination and light skin can appear dark in the opposite case. Moreover, there are many elements that influence the expression of accurate colors, such as thick makeup. The generator of pix2pix GAN could not determine the pixel values to the extent to accurately distinguish people.

Based on these results, this study used the data generated by pix2pix GAN and the original 81× augmented training data only when training the 1st CNN in Fig. 8. To train the 2nd and 3rd CNNs, only the original 81× augmented training data was used.

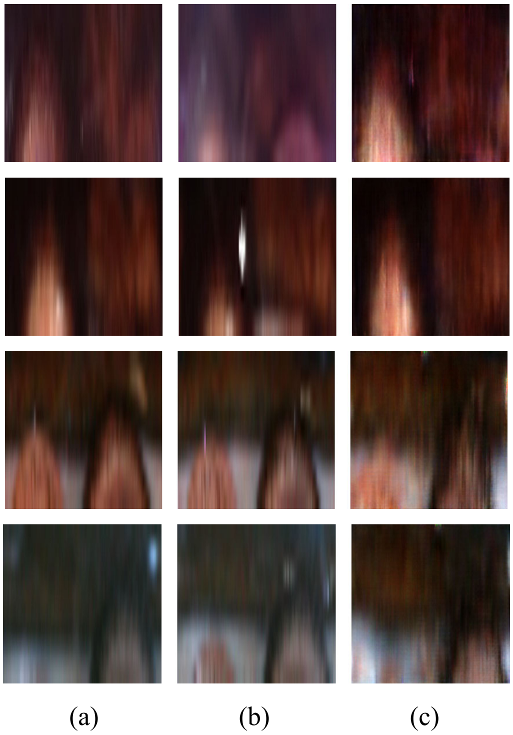


FIGURE 11. Examples of generated images by pix2pix GAN model with NICE.II training dataset. (a) Geometric center image, (b) input image, (c) generated output image by pix2pix GAN model. In (a) – (c), the images of the 1st and 2nd rows are the iris region of Fig. 3 (a), whereas those of the 3rd and 4th rows show the periocular regions based on $w_1 \times IRrad$ of Fig. 3, respectively.

In the subsequent experiment, a recognition accuracy comparison was made between the pix2pix GAN-based data augmentation proposed in this study and the geometric transform-based data augmentation used in previous studies [6], as shown in Table 6. The accuracy of the 1st CNN using iris region in Figure 8 was evaluated by two-fold cross validation, and the results showed that the recognition EER and d-prime value of the proposed method were better. Moreover, the score level fusion was performed by an SVM on the three Euclidean distances measured by the 4096-dimension features extracted from the three CNN models in Figure 8, and the results were an improvement over each CNN recognition result, as shown in Table 7. In addition, we compared the recognition errors by two methods. Method 1 is our method using iris region in polar coordinate generated by cGAN with the periocular region in polar coordinate generated by geometric transform. Method 2 uses whole eye or iris image in Cartesian coordinate generated by cGAN for training. For fair comparison, same SVM-based fusion was used in method 2, also. As shown in Table 7, our method based on polar coordinate shows lower error than that based on Cartesian coordinate. That is because the whole eye or iris region includes the various factors of skin, eyelid, and eyelashes, which hinder the correct generation by cGAN. However, iris region in polar coordinate does not include these factors and it is less affected by the arbitrary changes in the iris and pupil coordinates, which helps the correct generation by cGAN as shown in Figure 11.

TABLE 6. Comparison of recognition accuracies based on the proposed and previous methods.

Augmentation Method	Two-fold cross validation	EER (%)		d-prime value	
		Each fold	Avg.	Each fold	Avg.
Geometric transform-based [6]	1st fold	12.21	13.88	2.37	2.21
	2nd fold	15.54		2.04	
Pix2pix GAN-based (proposed method)	1st fold	12.07	12.79	2.23	2.25
	2nd fold	13.51		2.26	

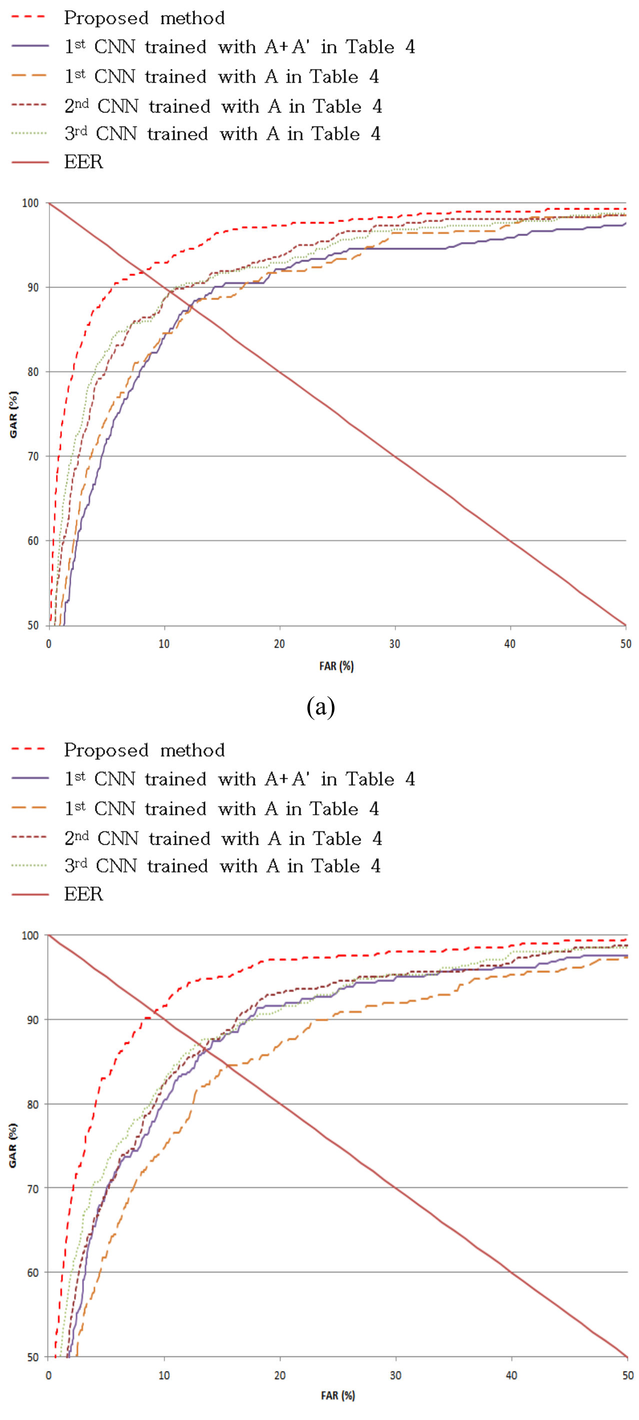
TABLE 7. Comparison of recognition accuracies based on the single CNN and three CNNs.

Method	Two-fold cross validation	EER (%)		d-prime value	
		Each fold	Average	Each fold	Average
Using one distance from the 1st CNN	1st fold	12.07	12.79	2.23	2.25
	2nd fold	13.51		2.26	
Using one distance from the 2nd CNN	1st fold	10.37	11.87	2.54	2.42
	2nd fold	13.36		2.29	
Using one distance from the 3rd CNN	1st fold	10.36	11.63	2.59	2.46
	2nd fold	12.89		2.32	
SVM classification (proposed method (Method 1))	1st fold	8.03	8.58	3.52	3.31
	2nd fold	9.12		3.09	
Method 2	1st fold	12.13	12.75	2.18	2.23
	2nd fold	13.37		2.28	

Figure 12 shows the ROC curves of the recognition accuracy. In Figure 12, the genuine acceptance rate (GAR) is 100 – FRR (%), and we can confirm that proposed method outperforms other methods. In Table 8, the recognition accuracy of the method proposed in this study is compared to that of the methods proposed in previous studies. As deduced from Table 8, the recognition accuracy of the method proposed in this study is better than that of the methods proposed in previous studies due to the training of CNN with the augmented data by cGAN. In the research [6], they also performed the training of CNN with the augmented data. However, the data were augmented by geometric transform, and these data have a high correlation with the original data. Therefore, they are unable to produce sufficient diversities.

E. TESTING OF IRIS RECOGNITION WITH MICHE DATABASE

Next, experiments were performed using the MICHE database. As described in Section V.*A* and Table 4, we performed a two-fold cross validation on each sub-dataset of the MICHE database. Table 9 compares the recognition accuracy of the pix2pix GAN-based data augmentation proposed in



(a)

(b)

FIGURE 12. ROC curves of recognition by proposed and other methods. (a) 1^{st} fold cross validation. (b) 2^{nd} fold cross validation.

TABLE 8. Comparison of recognition accuracies by proposed and previous methods (N.R means, "not reported").

Method	EER (%)	d-prime value
Sajjad et al. [49]	18.82	*N.R.*
Proença et al. [53]	16 (approximate value)	2.42 (approximate value)
Zanlorensi et al. [38]	13.98	2.25
Tan et al. [54]	12* (approximate value)	2.57
Lee et al. [6]	10.36	2.62
Proposed method	8.58	3.31

* : reported in the study by Proença et al. [53]

TABLE 9. Comparison of recognition accuracies based on the proposed and previous methods of data augmentation.

Method	Sub-dataset	Two-fold cross validation	EER (%)		d-prime value	
			Each fold	Avg.	Each fold	Avg.
Geometric transform-based [6]	Galaxy S4	1st fold	22.69	20.16	1.49	1.6
		2nd fold	17.63		1.7	
	Galaxy Tab2	1st fold	24.27	19.51	1.47	1.92
		2nd fold	14.75		2.37	
	iPhone5	1st fold	24.72	23.24	1.36	1.48
		2nd fold	21.76		1.59	
Pix2pix GAN-based (proposed method)	Galaxy S4	1st fold	19.58	18.08	1.61	1.72
		2nd fold	16.57		1.83	
	Galaxy Tab2	1st fold	22.4	19.24	1.78	1.98
		2nd fold	16.08		2.18	
	iPhone5	1st fold	20.88	19.5	1.57	1.67
		2nd fold	18.12		1.77	

TABLE 10. Recognition accuracies by the proposed method.

Sub-dataset	Two-fold cross validation	EER (%)		d-prime value	
		Each fold	Average	Each fold	Average
Galaxy S4	1st fold	18.16	15.49	1.86	2.09
	2nd fold	12.81		2.31	
Galaxy Tab2	1st fold	16.91	16.34	1.89	2.09
	2nd fold	15.77		2.29	
iPhone5	1st fold	17.74	17.39	1.8	1.87
	2nd fold	17.04		1.94	

this study and the geometric transform-based data augmentation used in previous studies [6]. The accuracy of the 1^{st} CNN in Fig. 8 was evaluated using the two-fold cross validation, and the results showed that the recognition EER and d-prime value of the method proposed in this study were better. Table 10 and Fig. 13 show the experimental results for the recognition method, which uses SVM-based score level fusion that was proposed in this paper. Subsequently, in Table 11, the recognition performance of our method and existing methods are compared. As seen in Table 11, the recognition performance of the method proposed in this study was better than that of the existing methods due to the training of CNN with the augmented data by cGAN. In the research [6], they also performed the training of CNN with the augmented data. However, the data were augmented by geometric transform, and these data have a high correlation with the original data. Therefore, they are unable to produce sufficient diversities.

F. TESTING OF IRIS RECOGNITION WITH CASIA-IRIS-DISTANCE DATABASE

To examine the possibility of applying the proposed method to a NIR camera and illumination images, the following

TABLE 11. Comparison of recognition accuracies by proposed and previous methods (*N.R.* means "not reported").

Method	Sub-dataset	EER (%)	d-prime value
Raja et al. [57]*	Galaxy S4	38.8	6.49
	Galaxy Tab2	33.9	8.63
	iPhone5	38.6	6.21
Santos et al. [58]*	Galaxy S4	19.8	6.13
	Galaxy Tab2	16.3	6.20
	iPhone5	22	5.44
Lee et al. [6]	Galaxy S4	17.9	1.87
	Galaxy Tab2	16.25	2.26
	iPhone5	17.45	2.00
Proposed method	Galaxy S4	15.49	2.09
	Galaxy Tab2	16.34	2.09
	iPhone5	17.39	1.87

* : accuracies are reported in the study by De Marsico et al. [17]

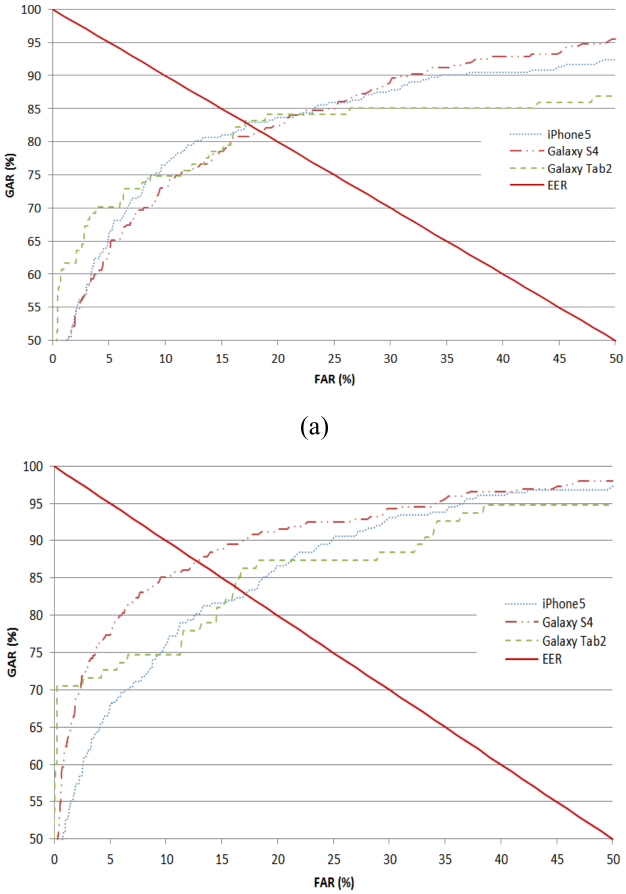
TABLE 12. Recognition accuracies by the proposed method.

Database	Two-fold cross validation	EER (%)		d-prime value	
		Each fold	Average	Each fold	Average
CASIA-Iris-Distance	1st fold	1.17	2.96	3.23	3.29
	2nd fold	4.75		3.35	

TABLE 13. Comparison of recognition accuracies by proposed and previous methods.

Method	EER (%)
Cho et al. [65]	10.02
Shekar et al. [66]	8.64
Wang et al. [64]	4.91
Zhao et al. [67]	4.9
Zhao et al. [31]	3.85
Sharifi et al. [63]	3.29
Lee et al. [6]	3.08
Proposed method	2.96

experiments were performed using the CASIA-Iris-Distance database. As described in Section V.*A* and Table 4, we performed a two-fold cross validation on each sub-dataset of CASIA-Iris-Distance database. Table 12 and Fig. 14 show the recognition accuracies by proposed method. Subsequently, the recognition performance of our method and existing methods are compared in Table 13. As seen in this table, the recognition performance of the method proposed in this study was better than that of the existing methods due to the training of CNN with the augmented data by cGAN. In the research [6], they also performed the training of CNN with the augmented data. However, the data were augmented by geometric transform, and these data have a high correlation with the original data. Therefore, they are unable to produce sufficient diversities.



(a)

(b)

FIGURE 13. ROC curves of recognition by proposed method. (a) 1st fold cross validation. (b) 2nd fold cross validation.

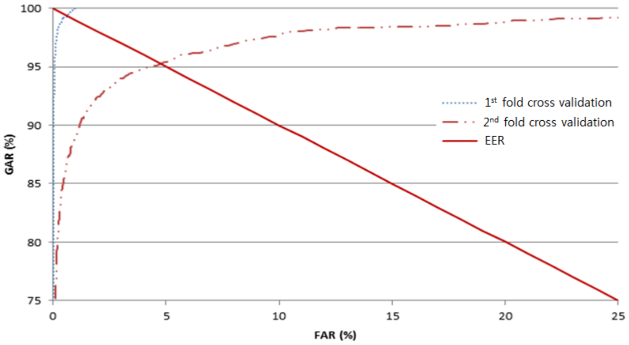


FIGURE 14. ROC curves of recognition by proposed method.

G. ANALYSIS OF EXPERIMENTAL RESULTS

Fig. 15 shows examples of genuine recognition successes using the proposed method. As shown in Figs. 15(a) and (b), the recognition was successful even though the two irides, for which recognition was being attempted, had different sizes and were at an off-angle. In Fig. 15(c), genuine recognition was successful there were extreme changes in the visible light illumination brightness. In Fig. 15(d), genuine recognition

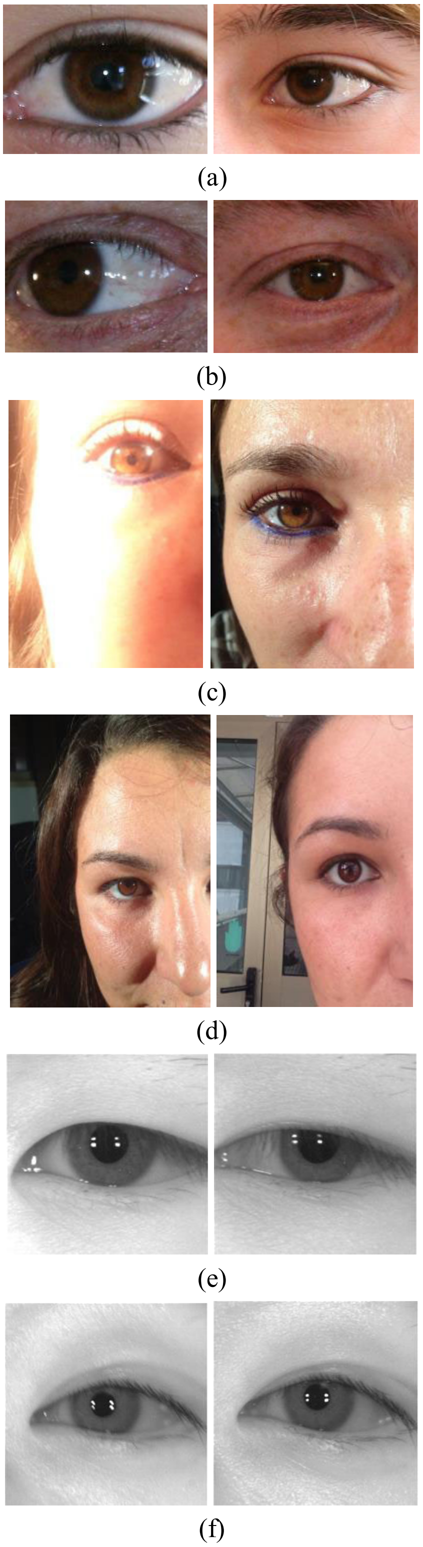


FIGURE 15. Recognition success cases (correct recognition cases of genuine matching). (a) Example of robustness with regard to change in iris size (NICE.II training database). (b) Example of robustness with regard to change in iris size and off-angle view (NICE.II training database). (c) Example of robustness with regard to severe illumination change (MICHE database). (d) Example of robustness with regard to severe reflected light in the iris (MICHE database). (e) Example of robustness with regard to occlusion by eyelashes and eyelid (CASIA-Iris-Distance database). (f) Example of robustness with regard to motion blurring (CASIA-Iris-Distance database).

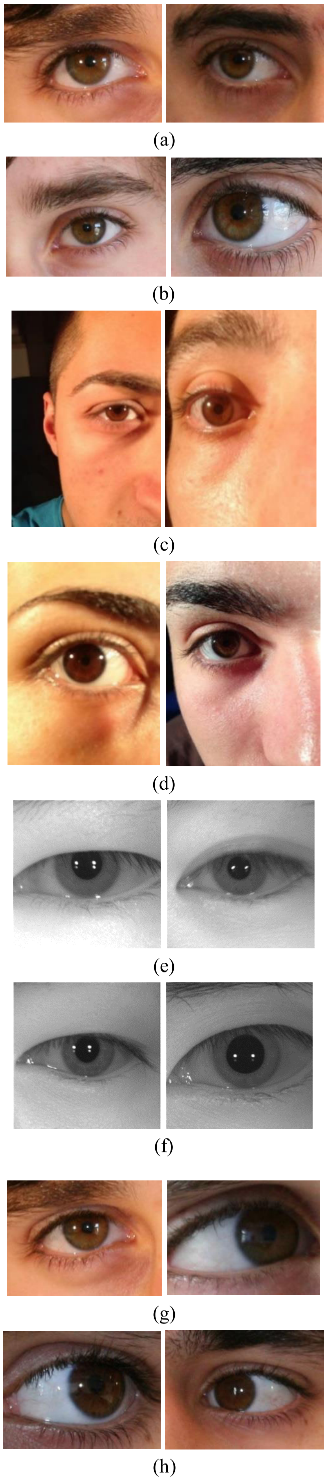


FIGURE 16. Recognition failure cases. (a) – (f) depicts false acceptance cases, whereas false rejection cases are shown in (g) – (h). (a) Error case due to unclear iris pattern, similar color, and wrinkle characteristics (NICE.II training database). (b) Error case due to noise resulting from an off-angle and reflected light, including great similarity in the wrinkles and skin color (NICE.II training database). (c) Error case due to similarity of the iris shape and periocular skin color (MICHE database). (d) Error case due to blurry iris and dark illumination (MICHE database). (e) Error case due to motion blurring and occlusion by eyelashes (CASIA-Iris-Distance database). (f) Error case due to change in iris size (CASIA-Iris-Distance database). (g) and (h) Examples of false rejection occurring due to off angle.

was successful even in the presence of severe reflected light in the iris, while the skin color and wrinkles appeared different due to changes in illumination. In Fig. 15(e), genuine recognition was successful even in the presence of severe occlusion by eyelashes and eyelid. In Fig. 15(f), the recognition was successful even though motion blurring occurred. Fig. 16 shows examples of recognition failure (false acceptance and rejection cases). Figs. 16(a) and (b) depict irides in different classes. However, because the irides, skin color, and wrinkles are very similar, and at an off-angle, they were recognized as the same person even though they are different people (false acceptance cases). Figs. 16(c) and (d) are different classes that were improperly recognized as the same person. This occurred because the shapes of the periocular areas, including wrinkles, etc. were very similar, while the iris patterns were not clear. Figs. 16(e) and (f) show the cases of false acceptance which occurs due to motion blurring, occlusion by eyelashes, or different iris size. The false rejection cases in Figs. 16(g) and (h) occurred due to off-angle. Overall, the method proposed in this study has strong advantages with regards to noise that commonly occurs in an unconstrained environment, such as changes in iris size, changes in visible light illumination, and off-angles. However, the method has a disadvantage, as it becomes highly dependent on the periocular region when the iris pattern is not clear, and the recognition rate drops in this case.

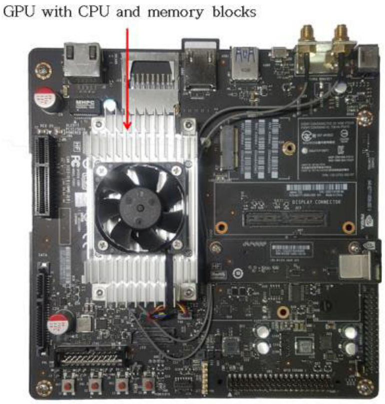


FIGURE 17. Jetson TX2 embedded system.

TABLE 14. Average processing time of proposed method (unit: ms).

	Desktop computer	Jetson TX2 embedded system
Obtain normalized images for the inputs to three CNN models	142	771
Obtain scores by three CNN models	40	147
Score fusion by SVM	24	40
Total	206	958

H. PROCESSING TIME OF PROPOSED METHOD

In Table 14, the average processing time per image was measured in the desktop environment described in Section 5.2. As shown in Table 14, the average processing time per image was 206 ms. The method proposed in this study was confirmed to perform processing at a rate of 4.85 frames per second. In the next experiment, the processing time was measured in the Jetson TX2 embedded system [61], shown in Fig. 17, which is a system that is already often used in onboard deep learning processing in self-driving cars. Jetson TX2 has a NVIDIA Pascal^{TM}-family GPU (256 CUDA cores), which has 8 GB of memory shared between the

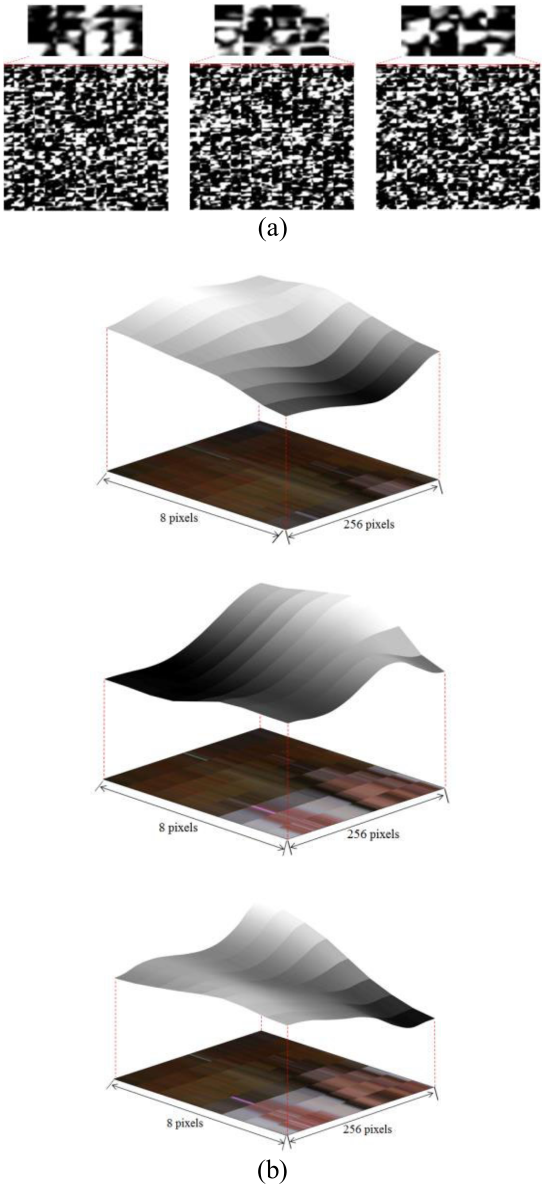


FIGURE 18. Examples of feature maps extracted from the last convolutional layer for the input images. In (a), the left, middle, and right FIGs depict the feature maps from the 1st, 2nd, and 3rd CNNs of Fig. 8, respectively, whereas in (b), the upper, middle, and lower FIGs show the feature maps from the 1st, 2nd, and 3rd CNNs of Fig. 8, respectively. Feature maps from (a) 8th convolutional layer. (b) 3-dimensional feature map image obtained by averaging all feature map values of (a).

central processing unit (CPU) and GPU, and 59.7 GB/s of memory bandwidth; it uses less than 7.5 watts of power. As shown in Table 14, the average processing time per image was 958 ms. Therefore, it was confirmed that the method proposed in this study can process at a rate of 1.04 frames per second. Compared to the desktop computer, the Jetson TX2 embedded system had a longer processing time due to its highly limited computing resources. Nevertheless, the method proposed in this study was confirmed to function even in embedded systems with limited computing resources.

I. ANALYSIS OF FEATURE MAP

Generally, the size of the feature map (width and height) becomes smaller as the depth of the convolutional layer increases, whereas the number of feature map channels increases. The closer the layer is to an input with a large image size, whereas the smaller the number of filters, and the farther the layer is from the input, the larger the number of filters. This sub-section analyzes the feature maps obtained from the three CNN models shown in Fig. 8, which were used in this study, as shown in Fig. 18.

Fig. 18(a) shows the feature maps of the 8th convolutional layer. As mentioned above, Fig. 18(b) shows a 3-dimensional feature map image obtained by finding the average of the feature map values of all channels in Fig. 18(a). As observed from the magnitudes of the feature map values in Fig. 18(b), the regions showing higher magnitudes of the feature map are different according to the 1st, 2nd, and 3rd CNNs of Fig. 8. Based on this observation, we can confirm that both iris and periocular regions provide useful features for recognition.

VI. CONCLUSION

This study has proposed a new iris recognition method, where CNN networks are trained with training data generated by the pix2pix GAN model, and recognition is performed. The iris and pupil coordinates were adjusted to normalize the iris images, and these images are entered as a training dataset in the pix2pix GAN, which has a cGAN structure in order to perform the training. The training dataset was input again as a testing dataset in the fully trained generator of the pix2pix GAN, and data augmentation was performed. The augmented dataset created by the generator and the augmented data generated by a geometric transform-based method were combined to train the CNN network, which uses the iris region as input. The two CNN networks that use the periocular region as input were trained using the augmented data generated by the existing geometric transform-based method. An SVM was used to perform score level fusion on the 3 distances, which are based on the three pairs of features obtained by each CNN, and authentic and imposter matching were performed. The NICE.II training dataset, MICHE database, and CASIA-Iris-Distance database were used, and performance was measured by a two-fold cross validation method. In the results, the method proposed in this study showed higher performance than the methods of previous studies. In addition, the processing speed of the proposed algorithm was measured on a desktop computer and a NVIDIA Jetson TX2 embedded system, and the results confirmed the ability to process at high speeds.

In future work, we plan to study methods for reducing errors related to imposter matching, which occur in environments where the iris pattern is not clear and dependence on the periocular region is high. In addition, we plan to study methods that can reduce various noise in the input images and improve recognition performance through a wider variety of GAN-based methods.

REFERENCES

[1] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," Jul. 2012, pp. 1–18, *arXiv:1207.0580*. [Online]. Available: https://arxiv.org/abs/1207.0580

[2] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Mach. Learn.*, Lille, France, Feb. 2015, pp. 448–456.

[3] A. Antoniou, A. Storkey, and H. Edwards, "Data augmentation generative adversarial networks," Nov. 2018, pp. 1–14, *arXiv:1711.04340*. [Online]. Available: https://arxiv.org/abs/1711.04340

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Miami, FL, USA, Jun. 2009, pp. 248–255.

[5] Y. Zhu, M. Aoun, M. Krijn, J. Vanschoren, and H. T. Campus, "Data augmentation using conditional generative adversarial networks for leaf counting in arabidopsis plants," in *Proc. Brit. Mach. Vis. Conf. Workshop Comput. Vis. Problems Plant Phenotyping*, Newcastle, U.K., Sep. 2018, pp. 1–11.

[6] M. B. Lee, H. G. Hong, and K. R. Park, "Noisy ocular recognition based on three convolutional neural networks," *Sensors*, vol. 17, no. 12, p. 2933, 2017.

[7] H. G. Hong, M. B. Lee, and K. R. Park, "Convolutional neural network-based finger-vein recognition using NIR image sensors," *Sensors*, vol. 17, no. 6, p. 1297, 2017.

[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. 28th Conf. Neural Inf. Process. Syst.*, Montreal, QC, Canada, Dec. 2014, pp. 1–9.

[9] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *Proc. Int. Conf. Learn. Represent.*, San Juan, PR, USA, Jan. 2016, pp. 1–16.

[10] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Honolulu, HI, USA, Jul. 2017, pp. 5967–5976.

[11] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, Munich, Germany, Oct. 2015, pp. 234–241.

[12] M. V. Giuffrida, H. Scharr, and S. A. Tsaftaris, "ARIGAN: Synthetic arabidopsis plants using generative adversarial network," in *Proc. IEEE Int. Conf. Comput. Vis.*, Venice, Italy, Oct. 2017, pp. 2064–2071.

[13] *NICE. II Training Dataset*. Accessed: Mar. 2, 2019. [Online]. Available: http://nice2.di.ubi.pt/

[14] M. Frid-Adar, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan, "Synthetic data augmentation using GAN for improved liver lesion classification," in *Proc. IEEE 15th Int. Symp. Biomed. Imag.*, Washington, DC, USA, Apr. 2018, pp. 289–293.

[15] S. Minaee and A. Abdolrashidi, "Finger-GAN: Generating realistic fingerprint images using connectivity imposed GAN," Dec. 2018, pp. 1–6, *arXiv:1812.10482*. [Online]. Available: https://arxiv.org/abs/1812.10482

[16] G. Wang, W. Kang, Q. Wu, Z. Wang, and J. Gao, "Generative adversarial network (GAN) based data augmentation for palmprint recognition," in *Proc. Digit. Image Comput., Techn. Appl.*, Canberra, Australia, Dec. 2018, pp. 1–7.

[17] M. De Marsico, M. Nappi, F. Narducci, and H. Proença, "Insights into the results of MICHE I—Mobile iris challenge evaluation," *Pattern Recognit.*, vol. 74, pp. 286–304, Feb. 2018.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Lake Tahoe, NV, USA vol. 2012, pp. 1097–1105.

[19] S. Minaee and A. Abdolrashidi, "Iris-GAN: Learning to generate realistic iris images using convolutional GAN," Dec. 2018, pp. 1–6, *arXiv:1812.04822*. [Online]. Available: https://arxiv.org/abs/1812.04822

[20] C. C. Chang and C. J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–27, 2011.

[21] *LIBSVM—A Library for Support Vector Machines*. Accessed: Mar. 22, 2019. [Online]. Available: https://www.csie.ntu.edu.tw/~cjlin/libsvm/

[22] D. T. Nguyen, T. D. Pham, Y. W. Lee, and K. R. Park, "Deep learning-based enhanced presentation attack detection for iris recognition by combining features from local and global regions based on NIR camera sensor," *Sensors*, vol. 18, no. 8, p. 2601, 2018.

[23] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," in *Proc. 33rd Int. Conf. Mach. Learn.*, New York, NY, USA, Jun. 2016, pp. 1558–1566.

[24] M. Mirza and S. Osindero, "Conditional generative adversarial nets," Nov. 2014, pp. 1–7, *arXiv:1411.1784*. [Online]. Available: https://arxiv.org/abs/1411.1784

[25] H. Proenca, S. Filipe, R. Santos, J. Oliveira, and L. A. Alexandre, "The UBIRIS.v2: A database of visible wavelength iris images captured on-the-move and at-a-distance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 8, pp. 1529–1535, Aug. 2010.

[26] *MICHE Database*. Accessed: Mar. 22, 2019. [Online]. Available: http://biplab.unisa.it/MICHE/database/

[27] B. Kaur, S. Singh, and J. Kumar, "Iris recognition using Zernike moments and polar harmonic transforms," *Arabian J. Sci. Eng.*, vol. 43, pp. 7209–7218, Dec. 2018.

[28] M. H. Hamd and S. K. Ahmed, "Fourier descriptors for iris recognition," *Int. J. Comput. Digit. Syst.*, vol. 6, no. 5, pp. 285–291, 2017.

[29] N. U. Ahmed, S. Cvetkovic, E. H. Siddiqi, A. Nikiforov, and I. Nikiforov, "Combining iris and periocular biometric for matching visible spectrum eye images," *Pattern Recognit. Lett.*, vol. 91, pp. 11–16, May 2017.

[30] C. Galdi and J.-L. Dugelay, "FIRE: Fast iris recognition on mobile phones by combining colour and texture features," *Pattern Recognit. Lett.*, vol. 91, pp. 44–51, May 2017.

[31] Z. Zhao and A. Kumar, "Towards more accurate iris recognition using deeply learned spatially corresponding features," in *Proc. IEEE Int. Conf. Comput. Vis.*, Venice, Italy, Oct. 2017, pp. 3829–3838.

[32] A. Gangwar and A. Joshi, "DeepIrisNet: Deep iris representation with applications in iris recognition and cross-sensor iris recognition," in *Proc. IEEE Int. Conf. Image Process.*, Phoenix, AZ, USA, Sep. 2016, pp. 2301–2305.

[33] M. G. Alaslani and L. A. Elrefaei, "Convolutional neural network-based feature extraction for iris recognition," *Int. J. Comput. Sci. Inf. Technol.*, vol. 10, pp. 65–78, Apr. 2018.

[34] S. Minaee, A. Abdolrashidiy, and Y. Wang, "An experimental study of deep convolutional features for iris recognition," in *Proc. IEEE Signal Process. Med. Biol. Symp.*, Philadelphia, PA, USA, Dec. 2016, pp. 1–6.

[35] K. Nguyen, C. Fookes, A. Ross, and S. Sridharan, "Iris recognition with off-the-shelf CNN features: A deep learning perspective," *IEEE Access*, vol. 6, pp. 18848–18855, 2017.

[36] Q. Zhang, H. Li, Z. Sun, and T. Tan, "Deep feature fusion for iris and periocular biometrics on mobile devices," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 11, pp. 2897–2912, Nov. 2018.

[37] Y. Xu, T.-C. Chuang, and S.-H. Lai, "Deep neural networks for accurate iris recognition," in *Proc. 4th IAPR Asian Conf. Pattern Recognit.*, Nanjing, China, Nov. 2017, pp. 664–669.

[38] L. A. Zanlorensi, E. Luz, R. Laroca, A. S. Britto, L. S. Oliveira, and D. Menotti, "The impact of preprocessing on deep representations for iris recognition on unconstrained environments," in *Proc. 31st SIBGRAPI Conf. Graph., Patterns Images*, Parana, Brazil, Oct./Nov. 2018, pp. 289–296.

[39] *CASIA-Iris-Distance*. Accessed: Mar. 22, 2019. [Online]. Available: http://www.cbsr.ia.ac.cn/china/Iris%20Databases%20CH.asp

[40] *TensorFlow*. Accessed: Mar. 22, 2019. [Online]. Available: https://www.tensorflow.org/

[41] *NVIDIA GeForce GTX 1080*. Accessed: Mar. 22, 2019. [Online]. Available: https://www.geforce.com/hardware/desktop-gpus/geforce-gtx-1080/specifications

[42] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, San Diego, CA, USA, Dec. 2014, pp. 1–15.

[43] Y. Jia. (2013). *CAFFE: An Open Source Convolutional Architecture for Fast Feature Embedding*. Accessed: Mar. 22, 2019. [Online]. Available: http://caffe.berkeleyvision.org/

[44] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis.*, Santiago, Chile, Dec. 2015, pp. 1026–1034.

[45] R. Szewczyk, K. Grabowski, M. Napieralska, W. Sankowski, M. Zubert, and A. Napieralski, "A reliable iris recognition algorithm based on reverse biorthogonal wavelet transform," *Pattern Recognit. Lett.*, vol. 33, pp. 1019–1026, Jun. 2012.

[46] P. Li and H. Ma, "Iris recognition in non-ideal imaging conditions," *Pattern Recognit. Lett.*, vol. 33, pp. 1012–1018, Jun. 2012.

[47] M. De Marsico, M. Nappi, and D. Riccio, "Noisy iris recognition integrated scheme," *Pattern Recognit. Lett.*, vol. 33, pp. 1006–1011, Jun. 2012.

[48] P. Li, X. Liu, and N. Zhao, "Weighted co-occurrence phase histogram for iris recognition," *Pattern Recognit. Lett.*, vol. 33, pp. 1000–1005, Jun. 2012.

[49] M. Sajjad, C.-W. Ahn, and J.-W. Jung, "Iris image enhancement for the recognition of non-ideal iris images," *KSII Trans. Internet Inf. Syst.*, vol. 10, no. 4, pp. 1904–1926, 2016.

[50] K. Y. Shin, G. P. Nam, D. S. Jeong, D. H. Cho, B. J. Kang, K. R. Park, and J. Kim, "New iris recognition method for noisy iris images," *Pattern Recognit. Lett.*, vol. 33, no. 8, pp. 991–999, 2012.

[51] G. Santos and E. Hoyle, "A fusion approach to unconstrained iris recognition," *Pattern Recognit. Lett.*, vol. 33, pp. 984–990, Sep. 2012.

[52] Q. Wang, X. Zhang, M. Li, X. Dong, Q. Zhou, and Y. Yin, "Adaboost and multi-orientation 2D Gabor-based noisy iris recognition," *Pattern Recognit. Lett.*, vol. 33, pp. 978–983, Jun. 2012.

[53] H. Proença and G. Santos, "Fusing color and shape descriptors in the recognition of degraded iris images acquired at visible wavelengths," *Comput. Vis. Image Understand.*, vol. 116, pp. 167–178, Feb. 2012.

[54] T. Tan, X. Zhang, Z. Sun, and H. Zhang, "Noisy iris image matching by using multiple cues," *Pattern Recognit. Lett.*, vol. 33, pp. 970–977, Jun. 2012.

[55] A. F. Abate, M. Frucci, C. Galdi, and D. Riccio, "BIRD: Watershed based iris detection for mobile devices," *Pattern Recognit. Lett.*, vol. 57, pp. 41–49, May 2015.

[56] S. Barra, A. Casanova, F. Narducci, and S. Ricciardi, "Ubiquitous iris recognition by means of mobile devices," *Pattern Recognit. Lett.*, vol. 57, pp. 66–73, May 2015.

[57] K. B. Raja, R. Raghavendra, V. K. Vemuri, and C. Busch, "Smartphone based visible iris recognition using deep sparse filtering," *Pattern Recognit. Lett.*, vol. 57, pp. 33–42, May 2015.

[58] G. Santos, E. Grancho, M. V. Bernardo, and P. T. Fiadeiro, "Fusing iris and periocular information for cross-sensor recognition," *Pattern Recognit. Lett.*, vol. 57, pp. 52–59, May 2015.

[59] *Dongguk cGAN-Based Iris Image Generation Model and Generated Images (DGIM&GI)*. Accessed: Apr. 22, 2019. [Online]. Available: http://dm.dgu.edu/link.html

[60] V. Vapnik, *Statistical Learning Theory*. New York, NY, USA: Wiley, 1998.

[61] *Jetson TX2 Module*. Accessed: Feb. 24, 2019. [Online]. Available: https://www.nvidia.com/en-us/autonomous-machines/embedded-systems-dev-kits-modules/

[62] K. Y. Shin, Y. G. Kim, and K. R. Park, "Enhanced iris recognition method based on multi-unit iris images," *Opt. Eng.*, vol. 52, Apr. 2013, Art. no. 047201.

[63] O. Sharifi and M. Eskandari, "Optimal face-iris multimodal fusion scheme," *Symmetry*, vol. 8, no. 6, p. 46, 2016.

[64] K. Wang and A. Kumar, "Toward more accurate iris recognition using dilated residual features," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 12, pp. 3233–3245, Dec. 2019.

[65] S. R. Cho, G. P. Nam, K. Y. Shin, D. T. Nguyen, T. D. Pham, E. C. Lee, and K. R. Park, "Periocular-based biometrics robust to eye rotation based on polar coordinates," *Multimedia Tools Appl.*, vol. 76, pp. 11177–11197, May 2017.

[66] B. H. Shekar and S. S. Bhat, "Iris recognition using partial sum of second order Taylor series expansion," in *Proc. 10th Indian Conf. Comput. Vis., Graph. Image Process.*, Guwahati, India, Dec. 2016, pp. 1–8.

[67] Z. Zhao and A. Kumar, "Improving periocular recognition by explicit attention to critical regions in deep neural network," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 12, pp. 2937–2952, Dec. 2018.

[68] J. Zhang, C. Lu, X. Li, H.-J. Kim, and J. Wang, "A full convolutional network based on DenseNet for remote sensing scene classification," *Math. Biosci. Eng.*, vol. 16, no. 5, pp. 3345–3367, 2019.

[69] J. Zhang, X. Jin, J. Sun, J. Wang, and K. Li, "Dual model learning combined with multiple feature selection for accurate visual tracking," *IEEE Access*, vol. 7, pp. 43956–43969, 2019.

[70] J. Zhang, Y. Wu, W. Feng, and J. Wang, "Spatially attentive visual tracking using multi-model adaptive response fusion," *IEEE Access*, vol. 7, pp. 83873–83887, 2019.

MIN BEOM LEE received the B.S. degree in information and telecommunication engineering from Dongyang Mirae University, Seoul, South Korea, in 2016. He is currently pursuing the joint M.S. and Ph.D. degrees in electronics and electrical engineering with Dongguk University. His research interests include biometrics and deep learning.

YU HWAN KIM received the B.S. degree from Pai Chai University, Daejeon, South Korea, in 2016, and the master's degree from Kyungpook National University, Daegu, South Korea, in 2019, both in electronics engineering. He is currently pursuing the Ph.D. degree in electronics and electrical engineering with Dongguk University. His research interests include biometrics and deep learning.

KANG RYOUNG PARK received the B.S. and M.S. degrees in electronics engineering and the Ph.D. degree in electrical and computer engineering from Yonsei University, Seoul, South Korea, in 1994, 1996, and 2000, respectively. He has been a Professor with the Division of Electronics and Electrical Engineering, Dongguk University, since March 2013. His research interests include image processing and biometrics. He supervised this research and helped the revision of draft of the original article.